Curry-de Bruijn-Howard for Justification Logic

Eduardo Bonelli

UNQ and CONICET, Argentina

Depto. de Matemática, Universidade de Brasília, 24 Oct. 2012

Topic index

1. Justification Logic
2. Hypothetical Justification Logic
3. History-Aware Computation
4. The Certifying Mobile Calculus
5. Conclusions and avenues for further research

Interpreting Int on the basis of proof

Kolmogorov 1932, Gödel 1933

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \hookrightarrow \; ? \; \hookrightarrow \textit{Classical proofs}$$

$$\begin{array}{ll} \mathbf{T} & \Box A \supset A \\ \mathbf{K} & \Box(A \supset B) \supset \Box A \supset \Box B \\ \mathbf{4} & \Box A \supset \Box\Box A \\ & \text{+ MP + Nec} \end{array}$$

Interpreting $\mathbf{Int}$ on the basis of proof

Kolmogorov 1932, Gödel 1933

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \hookrightarrow \, ? \, \hookrightarrow \textit{Classical proofs}$$

$\mathbf{T}$	$\Box A \supset A$
$\mathbf{K}$	$\Box(A \supset B) \supset \Box A \supset \Box B$
$\mathbf{4}$	$\Box A \supset \Box\Box A$
	+ MP + Nec

- Reading $\Box A$ as $\exists x.Proof(x, \ulcorner A \urcorner)$ problematic

  $\mathbf{S4}$ theorem $\Box(\neg\Box \perp)$ expresses $Con(\mathbf{PA})$ provable in $\mathbf{PA}$
- Observed by Gödel [Gödel:1933] who posed two problems:

Interpreting $\mathbf{Int}$ on the basis of proof

Kolmogorov 1932, Gödel 1933

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \hookrightarrow \; ? \; \hookrightarrow \textit{Classical proofs}$$

$\mathbf{T}$	$\Box A \supset A$
$\mathbf{K}$	$\Box(A \supset B) \supset \Box A \supset \Box B$
$\mathbf{4}$	$\Box A \supset \Box\Box A$
	+ MP + Nec

- Reading $\Box A$ as $\exists x.Proof(x, \ulcorner A \urcorner)$ problematic

  $\mathbf{S4}$ theorem $\Box(\neg\Box \bot)$ expresses $Con(\mathbf{PA})$ provable in $\mathbf{PA}$
- Observed by Gödel [Gödel:1933] who posed two problems:
  1. modal logic of formal provability predicate $\exists x.Proof(x, \ulcorner A \urcorner)$

Interpreting Int on the basis of proof

Kolmogorov 1932, Gödel 1933

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \hookrightarrow \ ? \ \hookrightarrow \textit{Classical proofs}$$

$\mathbf{T}$	$\Box A \supset A$
$\mathbf{K}$	$\Box(A \supset B) \supset \Box A \supset \Box B$
$\mathbf{4}$	$\Box A \supset \Box\Box A$
	+ MP + Nec

- Reading $\Box A$ as $\exists x.Proof(x, \ulcorner A \urcorner)$ problematic

  $\mathbf{S4}$ theorem $\Box(\neg\Box \bot)$ expresses $Con(\mathbf{PA})$ provable in $\mathbf{PA}$
- Observed by Gödel [Gödel:1933] who posed two problems:
  1. modal logic of formal provability predicate $\exists x.Proof(x, \ulcorner A \urcorner)$
  2. exact intended provability semantics for $\mathbf{S4}$

Interpreting $\mathbf{Int}$ on the basis of proof

Kolmogorov 1932, Gödel 1933

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \hookrightarrow \ ? \ \hookrightarrow \mathit{Classical\ proofs}$$

$\mathbf{T}$	$\Box A \supset A$
$\mathbf{K}$	$\Box(A \supset B) \supset \Box A \supset \Box B$
$\mathbf{4}$	$\Box A \supset \Box\Box A$
	+ MP + Nec

- Reading $\Box A$ as $\exists x.Proof(x, \ulcorner A \urcorner)$ problematic

  $\mathbf{S4}$ theorem $\Box(\neg\Box\perp)$ expresses $Con(\mathbf{PA})$ provable in $\mathbf{PA}$
- Observed by Gödel [Gödel:1933] who posed two problems:
  1. modal logic of formal provability predicate $\exists x.Proof(x, \ulcorner A \urcorner)$
  2. exact intended provability semantics for $\mathbf{S4}$
- Both have been addressed
  1. Solovay [Solovay:1976] (completeness of Löb's logic)

Interpreting $\mathbf{Int}$ on the basis of proof

Kolmogorov 1932, Gödel 1933

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \hookrightarrow \ ? \ \hookrightarrow \mathit{Classical\ proofs}$$

$\mathbf{T}$	$\Box A \supset A$
$\mathbf{K}$	$\Box(A \supset B) \supset \Box A \supset \Box B$
$\mathbf{4}$	$\Box A \supset \Box\Box A$
	+ MP + Nec

- Reading $\Box A$ as $\exists x.Proof(x, \ulcorner A \urcorner)$ problematic

  $\mathbf{S4}$ theorem $\Box(\neg\Box\perp)$ expresses $Con(\mathbf{PA})$ provable in $\mathbf{PA}$
- Observed by Gödel [Gödel:1933] who posed two problems:
  1. modal logic of formal provability predicate $\exists x.Proof(x, \ulcorner A \urcorner)$
  2. exact intended provability semantics for $\mathbf{S4}$
- Both have been addressed
  1. Solovay [Solovay:1976] (completeness of Löb's logic)
  2. Artemov [Artemov:1994] ($\mathbf{JL}$)

Provability Logic or Gödel-Löb Logic ($\mathbf{GL}$)

Problem 1: Modal logic of formal provability predicate $\exists x.Proof(x, \ulcorner A \urcorner)$

$\mathbf{K}$	$\Box(A \supset B) \supset \Box A \supset \Box B$
$\mathbf{L}$	$\Box(\Box A \supset A) \supset \Box A$
	+ MP + Nec
$\mathbf{4}$	$\Box A \supset \Box\Box A$ (derivable)

- Def. Modal proposition A always provable if A^* provable in $\mathbf{PA}$ for any arithmetical interpretation $\_^*$.
- [Solovay:1976] (arithmetical completeness of $\mathbf{GL}$): $\mathbf{GL}$ is sound and complete w.r.t. always provable propositions

Justification Logic (formerly, The Logic of Proofs)

Problem 2: Exact intended provability semantics for $\mathbf{S4}$ [Artemov:1994]

- In logics of knowledge we read $\Box A$ as "A is known"
- In $\mathbf{JL}$ we write $s{::}A$ and read "A is known for explicit reason s"

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \overbrace{\hookrightarrow}^{a} \mathbf{JL} \overbrace{\hookrightarrow}^{b} \mathit{Classical\ proofs}$$

a. Realization theorem

b. Arithmetical soundness and completeness

Justification Logic

$$
\begin{array}{rcll}
s,t & ::= & c \,|\, x \,|\, s\cdot t \,|\, !s \,|\, s+t & \text{proof polynomials} \\
A,B & ::= & P \,|\, A \supset B \,|\, s{::}A & \text{propositions}
\end{array}
$$

A0	Finite set of axiom schemes of classical logic
A1	$s{::}A \supset A$
A2	$s{::}(A \supset B) \supset (t{::}A \supset s \cdot t{::}B)$
A3	$s{::}A \supset\, !s{::}s{::}A$
A4	$s{::}A \supset s+t{::}A$, $t{::}A \supset s+t{::}A$
	+ MP
	+ Nec: A axiom **A0**-**A4**, and c proof constant, implies $\vdash c{::}A$

Constant specification: set $c_1{::}A_1, \ldots, c_n{::}A_n$ where A_i axiom **A0**-**A4**

Metatheory 1/2

- Deduction

$$\Gamma, A \vdash B \text{ implies } \Gamma \vdash A \supset B$$

- Lifing

$$\vec{s}{::}\Gamma, \Delta \vdash A \text{ implies } \exists t(\vec{x}, \vec{y}) \text{ s.t. } \vec{s}{::}\Gamma, \vec{y}{::}\Delta \vdash t(\vec{s}, \vec{y}){::}A$$

  - Internalization (Corollary)

$$\Delta \vdash A \text{ implies } \exists t(\vec{y}) \text{ s.t. } \vec{y}{::}\Delta \vdash t(\vec{y}){::}A$$

- DP [Krupski:2006]

$$\vdash s{::}A \vee t{::}B \text{ implies } \vdash s{::}A \text{ or } \vdash t{::}B$$

- Multiconclusion

$$\vdash s{::}A \wedge t{::}B \supset (s+t){::}A \wedge (s+t){::}B$$

Metatheory 2/2

Realization theorem

- Forgetful projection $\_^o$: replace $s{::}B$ with $\Box B$

$$\mathbf{JL} \vdash A \text{ implies } \mathbf{S4} \vdash A^o$$

- For the converse define a $\mathbf{JL}$-Realization $\_^r$ as
  - assignment of proof polynomials to all occurrences of $\Box$
  - normal if all negative occurrences of $\Box$ are realized as proof variables

Realization Theorem [Artemov:1994]

$\mathbf{S4} \vdash A$ implies $\mathbf{JL} \vdash A^r$, for some normal $\mathbf{JL}$-Realization $\_^r$

Metatheory 2/2

Realization Theorem continued

- The role of **A4** ($s{::}A \supset (s+t){::}A, t{::}A \supset (s+t){::}A$)

$$\begin{array}{l} A \vdash A \vee B \\ B \vdash A \vee B \\ \Box A \vdash \Box(A \vee B) \\ \Box B \vdash \Box(A \vee B) \\ \Box A \vee \Box B \vdash \Box(A \vee B) \end{array}$$

$A \vdash A \vee B$
$B \vdash A \vee B$
$x{::}A \vdash (a \cdot x){::}(A \vee B)$
$y :: B \vdash (b \cdot y) :: (A \vee B)$
... A4 used here
$x :: A \vdash (a \cdot x + b \cdot y) :: (A \vee B)$
$y :: B \vdash (a \cdot x + b \cdot y) :: (A \vee B)$
$x :: A \vee y :: B \vdash (a \cdot x + b \cdot y) :: (A \vee B)$

Metatheory 2/2

Realization Theorem continued

- Artemov's original proof was by induction on cut-free Gentzen derivation of $\mathbf{S4}$-theorem; yielded algorithm for decorating $\Box$s
- [Breshnev,Kuznets:2006]: Similar but produces proof polynomials of at most quadratic length
- Self-referentiality (i.e. propositions of the form $c{::}A(c)$) are required in order to realize all $\mathbf{S4}$-theorems (eg. $\neg\Box(R \wedge \neg\Box R)$) [Breshnev,Kuznets:2006]
- Use of **A4** can be dispensed with if we allow non-injective specification sets and non-normal realizations [Kuznets:2009, ArtemovBeklemishev:2004]
- Semantic proof of Realization Theorem [Fitting:2009]

Semantics 1/2

- Provability semantics [Artemov:1994]

$$\mathbf{JL}(CS) \vdash A \text{ iff } \mathbf{PA} \vdash A^* \text{ for any } CS\text{-interpretation } {}^*$$

CS-interpretation: arithmetical interpretation where each constant in CS is mapped to a provable formula in $\mathbf{PA}$

- Justification semantics [Mkrtychev:1997]

$M = (\mathcal{A}, \Vdash)$ where $\Vdash$ usual truth evaluations of prop. letters and $\mathcal{A}$ predicate that verifies

  - $\mathcal{A}(s, A \supset B)$ and $\mathcal{A}(t, A)$ implies $\mathcal{A}(s \cdot t, B)$
  - $\mathcal{A}(t, A)$ implies $\mathcal{A}(!t, A)$
  - $\mathcal{A}(s, A)$ or $\mathcal{A}(t, A)$ implies $\mathcal{A}(s + t, A)$

Truth of modal propositions

$$\Vdash s{::}A \text{ iff } \mathcal{A}(s, A) \text{ and } \Vdash A$$

Semantics 2/2

- Kripke-style semantics [Fitting:2003,2005]
  $(\mathcal{G}, \mathcal{R})$ frame
  $\mathcal{E}$ evidence function on $(\mathcal{G}, \mathcal{R})$ if for all proof polynomials s and t, for all formulas A and B , and for all $\Gamma, \Delta \in \mathcal{G}$
  1. Application $A \supset B \in \mathcal{E}(\Gamma, s)$ and $A \in \mathcal{E}(\Gamma, t)$ implies $B \in \mathcal{E}(\Gamma, s \cdot t)$;
  2. Monotonicity $\Gamma \mathcal{R} \Delta$ implies $\mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Delta, t)$;
  3. Proof Checker $A \in \mathcal{E}(\Gamma, t)$ implies $t{::}A \in \mathcal{E}(\Gamma, !t)$; and
  4. Sum $\mathcal{E}(\Gamma, s) \cup \mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Gamma, s + t)$

  $\mathcal{M} = (\mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V})$ is a weak $\mathbf{JL}$-model provided $(\mathcal{G}, \mathcal{R})$ is a frame with $\mathcal{R}$ reflexive and transitive, $\mathcal{E}$ is an evidence function on $(\mathcal{G}, \mathcal{R})$, and $\mathcal{V}$ is a mapping from propositional variables to subsets of $\mathcal{G}$.
- Categorical semantics [Lengyel:2009]

Quantification

- Over individuals
  - Example: $c(y)::(\forall x.A(x) \supset A(y))$, $u::\forall x.A(x) \supset (c(y) \cdot u)::A(y)$
  - Set of valid formulas not RE [Artemov,Sidon-Yavorskaya:2001]
- Over proof variables
  - Example: $\exists x.x : A \quad (= \Box A)$
  - Set of valid formulas not RE [Yavorsky:2002]
  - Another approach:
    - Completeness for a fragment of $\mathbf{JL}$+axioms for quantification [Fitting:2005]
    - Kripke-style semantics
    - Connection with arithmetic broken but could still be useful for CS

1 Justification Logic

2 Hypothetical Justification Logic

3 History-Aware Computation

4 The Certifying Mobile Calculus

5 Conclusions and avenues for further research

Programming idiom behind $\mathbf{JL}$?

- Develop proof theory of Hypothetical $\mathbf{JL}$
- Explore Curry-de Bruijn Howard correspondence
- Reflect hypothetical reasoning
- Logic aware of its proofs $\Rightarrow$ programs that coexist with type derivations in a unified setting
- We will restrict our attention to the minimal fragment without plus ($\mathbf{JL}^m$)

Natural Deduction for $\mathbf{JL}$ of [Artemov:1998]

$$\frac{s:(A \supset B) \quad t:A}{(s\cdot t):A} \qquad \frac{s:A}{A} \qquad \frac{s:A}{!s:s:A}$$

$$\frac{s:A}{(s+t):A} \qquad \frac{t:A}{(s+t):A} \qquad \frac{\overset{\mathcal{D}}{\mathbf{A}}}{c:\mathbf{A}}$$

$$\begin{array}{rcl} U(B(M)) & \rightarrow & M \\ U(C(M)) & \rightarrow & M \\ U(P(M,N)) & \rightarrow & U(M)\,U(N) \\ U(S_i(M)) & \rightarrow & U(M) \end{array}$$

Natural Deduction for $\mathbf{JL}$ of [Artemov:1998]

$$\frac{s:(A \supset B) \quad t:A}{(s \cdot t):A} \qquad \frac{s:A}{A} \qquad \frac{s:A}{!s:s:A}$$

$$\frac{s:A}{(s+t):A} \qquad \frac{t:A}{(s+t):A} \qquad \frac{\overset{\mathcal{D}}{\mathbf{A}}}{c:\mathbf{A}}$$

$$\begin{array}{rcl} U(B(M)) & \to & M \\ U(C(M)) & \to & M \\ U(P(M,N)) & \to & U(M)\,U(N) \\ U(S_i(M)) & \to & U(M) \end{array}$$

- We seek to reflect expressions (proof polynomials) encoding natural deduction proofs
- Well-behaved introduction/elimination inference schemes (Inversion Principle)

Hypothetical judgements

- Hypothetical judgements

$$\begin{array}{l} v_1 : A_1 \ \mathit{valid}, \ldots, v_n : A_n \ \mathit{valid} \ ; \\ a_1 : B_1 \ \mathit{true}, \ldots, a_m : B_m \ \mathit{true} \end{array} \quad \vdash \quad A \ \mathit{true} \mid s$$

or

$$\Delta; \Gamma \vdash A \mid s$$

- Rooted in similar analysis for $\mathbf{S4}$ [Pfenning,Davies:2001, (based on) Martin-Löf:1985]
- Context split allows for better behaved natural deduction presentation [Bierman, de Paiva]

Basic inference schemes of $\mathbf{S4}$ Modal Logic

$$\Delta; \Gamma \vdash A$$

$$\frac{}{\Delta; \Gamma, a : A, \Gamma' \vdash A}\ \mathsf{H} \qquad \frac{}{\Delta, v : A, \Delta'; \Gamma \vdash A}\ \mathsf{mH}$$

$$\frac{\Delta; \Gamma, a : A \vdash B}{\Delta; \Gamma \vdash A \supset B}\ \supset \mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \supset B \qquad \Delta; \Gamma \vdash A}{\Delta; \Gamma \vdash B}\ \supset \mathsf{E}$$

$$\frac{\Delta; \cdot \vdash A}{\Delta; \Gamma \vdash \Box A}\ \Box\mathsf{I}$$

$$\frac{\Delta; \Gamma \vdash \Box A \qquad \Delta, v : A; \Gamma \vdash C}{\Delta; \Gamma \vdash C}\ \Box\mathsf{E}$$

Basic inference schemes of $\mathbf{JL}^m$

$$\Delta; \Gamma \vdash A \mid s$$

$$\frac{}{\Delta; \Gamma, a : A, \Gamma' \vdash A \mid a}\ \mathsf{H} \qquad \frac{}{\Delta, v : A, \Delta'; \Gamma \vdash A \mid v}\ \mathsf{mH}$$

$$\frac{\Delta; \Gamma, a : A \vdash B \mid s}{\Delta; \Gamma \vdash A \supset B \mid \lambda a : A.s}\ \supset \mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \supset B \mid s \quad \Delta; \Gamma \vdash A \mid t}{\Delta; \Gamma \vdash B \mid s \cdot t}\ \supset \mathsf{E}$$

$$\frac{\Delta; \cdot \vdash A \mid s}{\Delta; \Gamma \vdash s{::}A \mid !s}\ \Box\mathsf{I}$$

$$\frac{\Delta; \Gamma \vdash r{::}A \mid s \quad \Delta, v : A; \Gamma \vdash C \mid t}{\Delta; \Gamma \vdash C^v_r \mid \mathsf{LET}\, v : A = s\, \mathsf{IN}\, t}\ \Box\mathsf{E}$$

Basic inference schemes of $\mathbf{JL}^m$

$$\Delta; \Gamma \vdash A \mid s$$

$$\frac{}{\Delta; \Gamma, a : A, \Gamma' \vdash A \mid a}\ \mathsf{H} \qquad \frac{}{\Delta, v : A, \Delta'; \Gamma \vdash A \mid v}\ \mathsf{mH}$$

$$\frac{\Delta; \Gamma, a : A \vdash B \mid s}{\Delta; \Gamma \vdash A \supset B \mid \lambda a : A.s}\ \supset \mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \supset B \mid s \quad \Delta; \Gamma \vdash A \mid t}{\Delta; \Gamma \vdash B \mid s \cdot t}\ \supset \mathsf{E}$$

$$\frac{\Delta; \cdot \vdash A \mid s}{\Delta; \Gamma \vdash s{::}A \mid !s}\ \Box\mathsf{I}$$

☜ Logical awareness/reflection

$$\frac{\Delta; \Gamma \vdash r{::}A \mid s \quad \Delta, v : A; \Gamma \vdash C \mid t}{\Delta; \Gamma \vdash C^v_r \mid \mathsf{LET}\, v : A = s\, \mathsf{IN}\, t}\ \Box\mathsf{E}$$

Sample derivation

$$
\dfrac{\dfrac{}{\cdot; a : s{::}A \vdash s{::}A \mid a}\,\mathsf{H} \qquad \dfrac{\dfrac{\dfrac{}{w : A; \cdot \vdash A \mid w}\,\mathsf{mH}}{w : A; \cdot \vdash w{::}A \mid !w}\,\Box\mathsf{I}}{w : A; a : s{::}A \vdash !w{::}w{::}A \mid !!w}\,\Box\mathsf{I}}{\dfrac{\cdot; a : s{::}A \vdash !s{::}s{::}A \mid \mathsf{LET}\, w : A = a \,\mathsf{IN}\, !!w}{\cdot; \cdot \vdash s{::}A \supset !s{::}s{::}A \mid \lambda a : s{::}A.\mathsf{LET}\, w : A = a\,\mathsf{IN}\, !!w}\supset\mathsf{I}}\,\Box\mathsf{E}
$$

Computational reading of a program of type $s : A$?

Via normalisation

- Compute value of type A and justify it (include some encoding of proof code s)

Computational reading of a program of type $s : A$?

Via normalisation

- Compute value of type A and justify it (include some encoding of proof code s)
- Logical awareness: computation will produce non-typable programs

$$\dfrac{\begin{array}{c}\vdots\; s\\ A\end{array}}{s : A}\,\Box I \quad\rightarrow\quad \dfrac{\begin{array}{c}\vdots\; \boxed{s'}\\ A\end{array}}{s : A}\,\Box I$$

Computational reading of a program of type $s : A$?

Via normalisation

- Compute value of type A and justify it (include some encoding of proof code s)
- Logical awareness: computation will produce non-typable programs

$$\dfrac{\begin{array}{c}\vdots\, s\\ A\end{array}}{s : A}\,\Box I \quad \to \quad \dfrac{\begin{array}{c}\vdots\, s'\\ A\end{array}}{s : A}\,\Box I$$

$$\dfrac{\dfrac{\dfrac{v : A; a : A \vdash A \mid a}{v : A; \cdot \vdash A \supset A \mid \lambda a : A.a}\supset \mathsf{I} \qquad v : A; \cdot \vdash A \mid v}{v : A; \cdot \vdash A \mid (\lambda a : A.a) \cdot v}\supset \mathsf{E}}{v : A; \cdot \vdash (\lambda a : A.a) \cdot v{::}A \mid !((\lambda a : A.a) \cdot v)}\,\Box\mathsf{I}$$

Computational reading of a program of type $s : A$?

Via normalisation

- Compute value of type A and justify it (include some encoding of proof code s)
- Logical awareness: computation will produce non-typable programs

$$\dfrac{\begin{array}{c}\vdots\ s\\ A\end{array}}{s : A}\,\Box I \quad \to \quad \dfrac{\begin{array}{c}\vdots\ \boxed{s'}\\ A\end{array}}{s : A}\,\Box I$$

$$\dfrac{\dfrac{\dfrac{v : A; a : A \vdash A \mid a}{v : A; \cdot \vdash A \supset A \mid \lambda a : A.a}\supset \mathsf{I} \qquad v : A; \cdot \vdash A \mid v}{v : A; \cdot \vdash A \mid (\lambda a : A.a) \cdot v}\supset \mathsf{E}}{v : A; \cdot \vdash (\lambda a : A.a) \cdot v{::}A \mid !((\lambda a : A.a) \cdot v)}\,\Box\mathsf{I}$$

$$\dfrac{v : A; \cdot \vdash A \mid \ v}{v : A; \cdot \vdash \ (\lambda a : A.a) \cdot v \ {::}A \mid !((\lambda a : A.a) \cdot v)}\,\Box\mathsf{I}$$

Computational reading of a program of type $s : A$?

Via normalisation

- Compute value of type A and justify it (include some encoding of proof code s)
- Logical awareness: computation will produce non-typable programs

$$\dfrac{\begin{matrix}\vdots\; s \\ A\end{matrix}}{s : A}\,\Box I \quad \to \quad \dfrac{\begin{matrix}\vdots\; s' \\ A\end{matrix}}{s : A}\,\Box I$$

$$\dfrac{\dfrac{\dfrac{v : A; a : A \vdash A \mid a}{v : A; \cdot \vdash A \supset A \mid \lambda a : A.a}\supset \mathsf{I} \qquad v : A; \cdot \vdash A \mid v}{v : A; \cdot \vdash A \mid (\lambda a : A.a) \cdot v}\supset \mathsf{E}}{v : A; \cdot \vdash (\lambda a : A.a) \cdot v :: A \mid !((\lambda a : A.a) \cdot v)}\,\Box\mathsf{I}$$

$$\dfrac{v : A; \cdot \vdash A \mid v}{v : A; \cdot \vdash (\lambda a : A.a) \cdot v : A \mid !((\lambda a : A.a) \cdot v)}\,\Box\mathsf{I}$$

Recovering Subject Reduction

$$\begin{array}{ll} & \text{Proofs reflected in object logic} \\ \& & \text{Reduction relates proofs} \\ \hline & \text{Proof code compatibility} \end{array}$$

Recovering Subject Reduction

$$\begin{array}{ll} & \text{Proofs reflected in object logic} \\ \& & \text{Reduction relates proofs} \\ \hline & \text{Proof code compatibility} \end{array}$$

$$\frac{\Delta;\Gamma \vdash A \mid s \quad \Delta;\Gamma \vdash \mathsf{Eq}(A,s,t) \mid e}{\Delta;\Gamma \vdash A \mid t}\ \mathsf{Eq}$$

Compatibility code constructors (sample)

$$\begin{array}{lll} e & ::= & \mathfrak{r}(N) \\ & \mid & \mathfrak{ba}([a:A]M,N) \mid \mathfrak{bb}([v:A]M,N) \\ & \mid & \mathfrak{ap}\mathfrak{C}(e,\mathfrak{r}(N)) \mid \mathfrak{le}\mathfrak{C}(e,[v:A]\mathfrak{r}(M)) \end{array}$$

Recovering Subject Reduction

$$\frac{\Delta; \Gamma \vdash A \mid s \quad \Delta; \Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash A \mid t} \ \mathsf{Eq}$$

Recovering Subject Reduction

$$\frac{\Delta;\Gamma \vdash A \mid s \quad \Delta;\Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta;\Gamma \vdash A \mid t}\ \mathsf{Eq}$$

Let $\Delta = v : A$ and $s = (\lambda a : A.a) \cdot v$ in

$$\frac{\dfrac{\dfrac{\Delta; a : A \vdash A \mid a}{\Delta; \cdot \vdash A \supset A \mid \lambda a : A.a}\ \supset \mathsf{I} \qquad \Delta; \cdot \vdash A \mid v}{\Delta; \cdot \vdash A \mid s}\ \supset \mathsf{E}}{\Delta; \cdot \vdash s{::}A \mid !s}\ \Box\mathsf{I}$$

Recovering Subject Reduction

$$\frac{\Delta;\Gamma \vdash A \mid s \quad \Delta;\Gamma \vdash \mathsf{Eq}(A,s,t) \mid e}{\Delta;\Gamma \vdash A \mid t}\,\mathsf{Eq}$$

Let $\Delta = v : A$ and $s = (\lambda a : A.a) \cdot v$ in

$$\dfrac{\dfrac{\Delta;\cdot \vdash A \mid v \qquad \begin{array}{c}\vdots\,\pi\\ \Delta;\cdot \vdash \mathsf{Eq}(A,v,s) \mid \mathfrak{ba}(a^A.a,v)\end{array}}{\Delta;\cdot \vdash A \mid s}\,\mathsf{Eq}}{\Delta;\cdot \vdash s{::}A \mid !s}\,\Box\mathsf{I}$$

Normalisation as Rewriting

$$(\lambda a : A.M) \cdot N \quad \xrightarrow{\beta} \quad \mathfrak{ba}([a:A]M, N) \blacktriangleright M^a_N$$

$$\text{LET } v : A = !^s N \text{ IN } M \quad \xrightarrow{\beta_\Box} \quad \mathfrak{bb}([v:A]M, N) \blacktriangleright (M^{v^\bullet}_N)^{v^\circ}_s$$

$$(e \blacktriangleright M) \cdot N \quad \xrightarrow{\blacktriangleright L} \quad \mathfrak{ap}\mathfrak{C}(e, \mathfrak{r}(N)) \blacktriangleright M \cdot N$$

$$\text{LET } v : A = e \blacktriangleright N \text{ IN } M \quad \xrightarrow{\blacktriangleright xtr} \quad \mathfrak{le}\mathfrak{C}(e, [v:A]\mathfrak{r}(M)) \blacktriangleright \text{LET } v : A = N \text{ IN } M$$

Define:

$$\xrightarrow{\beta\blacktriangleright} \stackrel{\text{def}}{=} \xrightarrow{\beta} \cup \xrightarrow{\beta_\Box} \cup \xrightarrow{\blacktriangleright L} \cup \xrightarrow{\blacktriangleright xtr}$$

Properties of $\xrightarrow{\beta\blacktriangleright}$

- Can be encoded as orthog., higher-order, pattern rewrite system

> All orthogonal higher-order rewrite systems are confluent [TERESE:2003]

- Encoding yields non-erasing and fully-extended system

> Non-erasing, orthogonal and fully-extended second-order rewrite systems are uniformly normalising [Khasidashvili,Ogawa,vanOostrom:2001]

  - A rewrite system is uniformly normalising if all its steps preserve the possibility of infinite reductions.
  - SN follows hence from WN

Sample reduction

$$I \cdot (I \cdot b)$$

where $I = \lambda a : A.a$
Two different (cofinal!) reductions in standard lambda calculus

$$I \cdot (I \cdot b) \xrightarrow{\beta} I \cdot b$$
$$I \cdot (I \cdot b) \xrightarrow{\beta} I \cdot b$$

"Syntactic accident" [Lévy:1978]

Sample reduction

Reductions in $\stackrel{\beta\blacktriangleright}{\to}$ (no longer cofinal)

$$I \cdot (I \cdot b) \stackrel{\beta}{\to} I \cdot (\mathfrak{ba}([a:A]a, b) \blacktriangleright b)$$
$$I \cdot (I \cdot b) \stackrel{\beta}{\to} \mathfrak{ba}([a:A]a, (I \cdot b)) \blacktriangleright I \cdot b$$

What does confluence of $\stackrel{\blacktriangleright}{\to}$ say about $\stackrel{\beta}{\to}$?

- All normalizing derivations from a term M are Lévy permutation equivalent
- Not new [Lévy:1978,TERESE:2003], but labels (i.e. evidence) justified logically

1 Justification Logic

2 Hypothetical Justification Logic

3 History-Aware Computation

4 The Certifying Mobile Calculus

5 Conclusions and avenues for further research

Revisiting the introduction scheme for the Modality

- Eq permutes past all inference schemes...except for $\Box$I, suggesting:

$$\frac{\Delta \vdash A \mid s}{\Delta; \Gamma \vdash s{::}A \mid !s} \Box\mathsf{I}$$

Revisiting the introduction scheme for the Modality

- Eq permutes past all inference schemes...except for $\Box$I, suggesting:

$$\frac{\Delta \vdash A \mid s \quad \Delta; \cdot \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash t{::}A \mid !t}\ \Box\mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \mid s \quad \Delta; \Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash A \mid t}\ \mathsf{Eq}$$

- If e is interpreted as the computation history or trail, then $\Box$I may be seen to
  - Introduce audited computation units
  - Trails are locally scoped

Revisiting the introduction scheme for the Modality

$$\frac{\Delta \vdash A \mid s \quad \Delta; \cdot \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash t{::}A \mid !t}\ \Box\mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \mid s \quad \Delta; \Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash A \mid t}\ \mathsf{Eq}$$

Revisiting the introduction scheme for the Modality

$$\frac{\Delta \vdash A \mid s \quad \Delta; \cdot \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash t{::}A \mid !t} \Box\mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \mid s \quad \Delta; \Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash A \mid t} \mathsf{Eq}$$

Let $\Delta = v : A$ and $s = (\lambda a : A.a) \cdot v$ in

$$\frac{\dfrac{\dfrac{\Delta; a : A \vdash A \mid a}{\Delta; \cdot \vdash A \supset A \mid \lambda a : A.a} \supset \mathsf{I} \qquad \Delta; \cdot \vdash A \mid v}{\Delta; \cdot \vdash A \mid s} \supset \mathsf{E} \qquad \begin{array}{c} \vdots \, \pi_1 \\ \Delta; \cdot \vdash \mathsf{Eq}(A, s, s) \mid \mathfrak{r}(s) \end{array}}{\Delta; \cdot \vdash s{::}A \mid !s} \Box\mathsf{I}$$

Revisiting the introduction scheme for the Modality

$$\frac{\Delta \vdash A \mid s \quad \Delta; \cdot \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash t{::}A \mid !t} \ \Box\mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \mid s \quad \Delta; \Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash A \mid t} \ \mathsf{Eq}$$

Let $\Delta = v : A$ and $s = (\lambda a : A.a) \cdot v$ in

$$\cfrac{\cfrac{\Delta; \cdot \vdash A \mid \qquad \begin{array}{c} \vdots\ \pi_2 \\ \Delta; \cdot \vdash \mathsf{Eq}(A, v, s) \mid \mathfrak{ba}(a^A.a, v) \end{array}}{\Delta; \cdot \vdash A \mid s} \ \mathsf{Eq} \qquad \begin{array}{c} \vdots\ \pi_1 \\ \Delta; \cdot \vdash \mathsf{Eq}(A, s, s) \mid \mathfrak{r}(s) \end{array}}{\Delta; \cdot \vdash s{::}A \mid !s} \ \Box\mathsf{I}$$

Revisiting the introduction scheme for the Modality

$$\frac{\Delta \vdash A \mid s \quad \Delta; \cdot \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash t{::}A \mid !t}\ \Box\mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \mid s \quad \Delta; \Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash A \mid t}\ \mathsf{Eq}$$

Let $\Delta = v : A$ and $s = (\lambda a : A.a) \cdot v$ in

$$\frac{\dfrac{\Delta; \cdot \vdash A \mid v \qquad \overset{\vdots\ \pi_2}{\Delta; \cdot \vdash \mathsf{Eq}(A, v, s) \mid \mathfrak{ba}(a^A.a, v)}}{\Delta; \cdot \vdash A \mid s}\ \mathsf{Eq} \qquad \overset{\vdots\ \pi_1}{\Delta; \cdot \vdash \mathsf{Eq}(A, s, s) \mid \mathfrak{r}(s)}}{\Delta; \cdot \vdash s{::}A \mid !s}\ \Box\mathsf{I}$$

Revisiting the introduction scheme for the Modality

$$\frac{\Delta \vdash A \mid s \quad \Delta; \cdot \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash t{::}A \mid !t} \Box\mathsf{I} \qquad \frac{\Delta; \Gamma \vdash A \mid s \quad \Delta; \Gamma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma \vdash A \mid t} \mathsf{Eq}$$

Let $\Delta = v : A$ and $s = (\lambda a : A.a) \cdot v$ in

$$\frac{\Delta; \cdot \vdash A \mid v \qquad \begin{array}{c} \vdots \;\; \pi_2 \circ \pi_1 \\ \Delta; \cdot \vdash \mathsf{Eq}(A, v, s) \mid \mathfrak{t}(\mathfrak{ba}(a^A.a, v), \mathfrak{r}(s)) \end{array}}{\Delta; \cdot \vdash s{::}A \mid !s} \Box\mathsf{I}$$

Trail access

Reaching out to programming languages

$$\dfrac{\begin{array}{c}\vdots\, s' \\ A\end{array} \qquad \begin{array}{c}\vdots\, e \\ Eq(A, s', s)\end{array}}{s : A}\ \Box I$$

Trail access

Reaching out to programming languages

$$
\dfrac{\begin{array}{c}\theta^B\alpha : B \\ \vdots\ s' \\ A\end{array} \qquad \begin{array}{c}\vdots\ e \\ Eq(A, s', s)\end{array}}{s : A}\ \Box I
$$

Trail access

Reaching out to programming languages

$$
\dfrac{\begin{array}{c}\theta^B\alpha : B \\ \vdots\ s' \\ A\end{array} \qquad \begin{array}{c}\vdots\ e \\ Eq(A, s', s)\end{array}}{s : A}\ \Box I
\quad \rightarrow \quad
\dfrac{\begin{array}{c}\vdots\ \theta^B e \\ B \\ \vdots\ s'' \\ A\end{array} \qquad \begin{array}{c}\vdots\ e' \\ Eq(A, s'', s)\end{array}}{s : A}\ \Box I
$$

Trail access

Reaching out to programming languages

$$\dfrac{\begin{array}{c}\theta^B\alpha : B\\ \vdots\; s'\\ A\end{array} \qquad \begin{array}{c}\vdots\; e\\ Eq(A, s', s)\end{array}}{s : A}\,\Box I \quad \rightarrow \quad \dfrac{\begin{array}{c}\vdots\; \boxed{\theta^B e}\\ B\\ \vdots\; s''\\ A\end{array} \qquad \begin{array}{c}\vdots\; e'\\ Eq(A, s'', s)\end{array}}{s : A}\,\Box I$$

- $\theta^B e$ traverses e to obtain term of type B (eg. count β steps; $B =$ Nat)

Trail access

Reaching out to programming languages

$$\frac{\begin{array}{c}\theta^B\alpha : B\\ \vdots\ s'\\ A\end{array} \qquad \begin{array}{c}\vdots\ e\\ Eq(A, s', s)\end{array}}{s : A}\ \Box I \quad \to \quad \frac{\begin{array}{c}\vdots\ \boxed{\theta^B e}\\ B\\ \vdots\ s''\\ A\end{array} \qquad \begin{array}{c}\vdots\ e'\\ Eq(A, s'', s)\end{array}}{s : A}\ \Box I$$

- $\theta^B e$ traverses e to obtain term of type B (eg. count β steps; $B = \mathsf{Nat}$)
- Trail variables are affine in nature

Trail access

Reaching out to programming languages

$$\dfrac{\begin{array}{c}\theta^B\alpha : B\\ \vdots\ s'\\ A\end{array} \qquad \begin{array}{c}\vdots\ e\\ Eq(A, s', s)\end{array}}{s : A}\,\Box I \quad \to \quad \dfrac{\begin{array}{c}\vdots\ \boxed{\theta^B e}\\ B\\ \vdots\ s''\\ A\end{array} \qquad \begin{array}{c}\vdots\ e'\\ Eq(A, s'', s)\end{array}}{s : A}\,\Box I$$

- $\theta^B e$ traverses e to obtain term of type B (eg. count β steps; $B = \text{Nat}$)
- Trail variables are affine in nature
- Final form for $\Box I$

$$\dfrac{\Delta; \cdot; \Sigma \vdash A \mid s \qquad \Delta; \cdot; \Sigma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma; \Sigma' \vdash \Sigma.t :: A \mid \Sigma.t}\,\Box\mathsf{I}$$

Inference schemes – summary

$$\Delta; \Gamma; \Sigma \vdash A \mid s$$

$$\frac{a : A \in \Gamma}{\Delta; \Gamma; \Sigma \vdash A \mid a}\ \mathsf{H} \qquad \frac{u : A[\Sigma] \in \Delta \quad \Sigma\sigma \subseteq \Sigma'}{\Delta; \Gamma; \Sigma' \vdash A \mid \langle u; \sigma \rangle}\ \mathsf{mH}$$

$$\frac{\Delta; \Gamma_1; \Sigma_1 \vdash A \supset B \mid s \quad \Delta; \Gamma_2; \Sigma_2 \vdash A \mid t}{\Delta; \Gamma_{1,2}; \Sigma_{1,2} \vdash B \mid s \cdot t}\ \supset \mathsf{E} \qquad \frac{\Delta; \Gamma, a : A; \Sigma \vdash B \mid s}{\Delta; \Gamma; \Sigma \vdash A \supset B \mid \lambda a : A.s}\ \supset \mathsf{I}$$

$$\frac{\Delta; \cdot; \Sigma \vdash A \mid s \quad \Delta; \cdot; \Sigma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma; \Sigma' \vdash \Sigma.t{::}A \mid \Sigma.t}\ \Box\mathsf{I} \qquad \frac{\Delta; \Gamma_1; \Sigma_1 \vdash \Sigma.r{::}A \mid s \quad \Delta, u : A[\Sigma]; \Gamma_2; \Sigma_2 \vdash C \mid t}{\Delta; \Gamma_{1,2}; \Sigma_{1,2} \vdash C^u_{\Sigma.r} \mid \mathsf{LET}(u : A[\Sigma].t, s)}\ \Box\mathsf{E}$$

$$\frac{\alpha : \mathsf{Eq}(A) \in \Sigma \quad \Delta; \cdot; \cdot \vdash \mathcal{T}^B \mid \theta^B}{\Delta; \Gamma; \Sigma \vdash B \mid \alpha\theta^B}\ \mathsf{TI} \qquad \frac{\Delta; \Gamma; \Sigma \vdash A \mid s \quad \Delta; \Gamma; \Sigma \vdash \mathsf{Eq}(A, s, t) \mid e}{\Delta; \Gamma; \Sigma \vdash A \mid t}\ \mathsf{E}$$

Term assignment for $\mathbf{JL}$ – $\lambda^{\mathcal{H}}$

$$M \quad ::= \quad a \mid \lambda a : A.M \mid M\, M \qquad \text{Standard}$$

Term assignment for $\mathbf{JL}$ – $\lambda^{\mathcal{H}}$

$$
\begin{array}{rcll}
M & ::= & \ldots & \text{Standard} \\
 & | & !^{\Sigma}_{e} M & \text{Audited computation unit} \\
 & | & \langle u; \sigma \rangle & \text{Audited unit variable} \\
 & | & \mathsf{let}\, u : A[\Sigma] = M \,\mathsf{in}\, M & \text{Audited unit composition} \\
 & | & \alpha\theta^{B} & \text{Audit trail lookup} \\
 & | & e \rhd M & \text{Partial trail}
\end{array}
$$

Term assignment for $\mathbf{JL}$ – $\lambda^{\mathcal{H}}$

M	$::=$	$\ldots$	Standard
	$\mid$	$!^{\Sigma}_{e} M$	Audited computation unit
	$\mid$	$\langle u; \sigma \rangle$	Audited unit variable
	$\mid$	let $u : A[\Sigma] = M$ in M	Audited unit composition
	$\mid$	$\alpha\theta^{B}$	Audit trail lookup
	$\mid$	$e \rhd M$	Partial trail

$!^{\alpha}_{e}\ \mathtt{if}\ \alpha\theta^{\mathbb{N}} > \underline{5}\ \mathtt{then}\ \underline{1}\ \mathtt{else}\ \underline{2}$

“If more than five β steps have been applied in the audited unit, then return 1 else return 2”

$$
\begin{array}{rcl}
\theta^{\mathbb{N}}(\beta) & \stackrel{\text{def}}{=} & \underline{1} \\
\theta^{\mathbb{N}}(\beta_{\Box}) & \stackrel{\text{def}}{=} & \underline{0} \\
\theta^{\mathbb{N}}(\mathfrak{r}) & \stackrel{\text{def}}{=} & \underline{0} \\
\theta^{\mathbb{N}}(\mathfrak{s}) & \stackrel{\text{def}}{=} & \lambda a : \mathbb{N}.a \\
\theta^{\mathbb{N}}(\mathfrak{t}) & \stackrel{\text{def}}{=} & \lambda a : \mathbb{N}.\lambda b : \mathbb{N}.a + b \\
\theta^{\mathbb{N}}(\mathfrak{App}) & \stackrel{\text{def}}{=} & \lambda a : \mathbb{N}.\lambda b : \mathbb{N}.a + b \\
 & & \ldots
\end{array}
$$

Operational Semantics of $\lambda^{\mathcal{H}}$ – CBV

Typed term reduction

$$
\begin{array}{rcll}
V & ::= & a \mid \langle u;\sigma\rangle \mid \lambda a:A.M \mid !^{\Sigma}_{e} V & \text{Values} \\
\theta^{B}_{V} & ::= & \{c_1/V_1,\ldots,c_{10}/V_{10}\} & \\
\mathcal{E} & ::= & \Box \mid \mathcal{E}\ M \mid V\ \mathcal{E} \mid \mathsf{let}\ u : A[\Sigma] = \mathcal{E}\ \mathsf{in}\ M & \text{Evaluation contexts} \\
 & \mid & !^{\Sigma}_{e}\mathcal{E} \mid \alpha\{c_1/V_1,\ldots,c_j/V_j,c_{j+1}/\mathcal{E},\ldots\} & \\
\mathcal{F} & ::= & \Box \mid \mathcal{F}\ M \mid V\ \mathcal{F} \mid \mathsf{let}\ u : [\Sigma] = \mathcal{F}\ \mathsf{in}\ M & \text{Lookup contexts}
\end{array}
$$

$$
\mathcal{E}[M] \mapsto \mathcal{E}[N] \quad \text{if} \quad M \rightharpoonup N \text{ where}
$$

$$
\begin{array}{rcl}
(\lambda a:A.M)\ V & \rightharpoonup_{\beta} & \mathfrak{ba}(a^{A}.s,t) \rhd M^{a}_{V,t} \\
\mathsf{let}\ u : A[\Sigma] = !^{\Sigma}_{e} V\ \mathsf{in}\ N & \rightharpoonup_{\beta_{\Box}} & \mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s) \rhd N^{u}_{\Sigma.(V,s,e)} \\
!^{\Sigma}_{e}\mathcal{F}[\alpha\theta^{B}_{V}] & \rightharpoonup_{\mathcal{L}} & !^{\Sigma}_{e}\mathcal{F}[\mathit{Trl}(\theta^{w}_{V},\alpha) \rhd e\theta_{V}]
\end{array}
$$

Operational Semantics of $\lambda^{\mathcal{H}}$ – CBV

Typed term reduction

$$
\begin{array}{llll}
V & ::= & a \mid \langle u;\sigma\rangle \mid \lambda a:A.M \mid !^{\Sigma}_{e} V & \text{Values} \\
\theta^{B}_{V} & ::= & \{c_1/V_1,\ldots,c_{10}/V_{10}\} & \\
\mathcal{E} & ::= & \Box \mid \mathcal{E}\ M \mid V\ \mathcal{E} \mid \mathsf{let}\ u:A[\Sigma] = \mathcal{E}\ \mathsf{in}\ M & \text{Evaluation contexts} \\
 & \mid & !^{\Sigma}_{e}\mathcal{E} \mid \alpha\{c_1/V_1,\ldots,c_j/V_j,c_{j+1}/\mathcal{E},\ldots\} & \\
\mathcal{F} & ::= & \Box \mid \mathcal{F}\ M \mid V\ \mathcal{F} \mid \mathsf{let}\ u:[\Sigma] = \mathcal{F}\ \mathsf{in}\ M & \text{Lookup contexts}
\end{array}
$$

$$\mathcal{E}[M] \mapsto \mathcal{E}[N] \quad \text{if} \quad M \rightharpoonup N \text{ where}$$

$$
\begin{array}{rcl}
(\lambda a:A.M)\ V & \rightharpoonup_{\beta} & \mathfrak{ba}(a^{A}.s,t) \rhd M^{a}_{V,t} \\
\mathsf{let}\ u:A[\Sigma] = !^{\Sigma}_{e} V\ \mathsf{in}\ N & \rightharpoonup_{\beta_{\Box}} & \mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s) \rhd N^{u}_{\Sigma.(V,s,e)} \\
!^{\Sigma}_{e}\mathcal{F}[\alpha\theta^{B}_{V}] & \rightharpoonup_{\mathcal{L}} & !^{\Sigma}_{e}\mathcal{F}[\mathit{Trl}(\theta^{w}_{V},\alpha) \rhd e\theta_{V}]
\end{array}
$$

Trail normalisation: occurrences of $e \rhd M$ are further normalised by shifting e towards its enclosing audited unit (reduction schemes omitted)

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)}\ \mathsf{let}\ u =!^{\alpha}_{e_1} P\ \mathsf{in}\ !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\ \underline{1}$$

where $P \stackrel{\text{def}}{=} \lambda a : \mathbb{N}.\texttt{if}\ \alpha\theta > \underline{5}\ \texttt{then}\ \texttt{error}\ \texttt{else}\ a$.

$$!_{\mathfrak{r}(r)}\ \mathsf{let}\ u =!^{\alpha}_{e_1} P\ \mathsf{in}\ !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\ \underline{1}$$

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)}\ \mathsf{let}\, u = !^{\alpha}_{e_1} P \,\mathsf{in}\, !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\, \underline{1}$$

where $P \stackrel{\mathrm{def}}{=} \lambda a : \mathbb{N}.\mathtt{if}\ \alpha\theta > \underline{5}\ \mathtt{then}\ \mathtt{error}\ \mathtt{else}\ a$.

$$\begin{array}{ll} & !_{\mathfrak{r}(r)}\ \mathsf{let}\, u = !^{\alpha}_{e_1} P \,\mathsf{in}\, !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\, \underline{1} \\ \mapsto_{\beta_\Box} & !_{\mathfrak{t}(\mathfrak{b}\mathfrak{b}(u^{A[\Sigma]}.t, \Sigma.s), \mathfrak{r}(r))}\ !^{\gamma}_{\mathfrak{r}(t)} (e_1 \rhd P)\{\alpha/\gamma\}\, \underline{1} \end{array}$$

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)} \text{ let } u = !^{\alpha}_{e_1} P \text{ in } !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle \, \underline{1}$$

where $P \stackrel{\text{def}}{=} \lambda a : \mathbb{N}.\texttt{if } \alpha\theta > \underline{5} \texttt{ then error else } a$.

$$!_{\mathfrak{r}(r)} \text{ let } u = !^{\alpha}_{e_1} P \text{ in } !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle \, \underline{1}$$

$$\mapsto_{\beta_\Box} \, !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t, \Sigma.s), \mathfrak{r}(r))} \, !^{\gamma}_{\mathfrak{r}(t)} (e_1 \rhd P)\{\alpha/\gamma\} \, \underline{1}$$

Trail persistence

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)} \text{ let } u =!^{\alpha}_{e_1} P \text{ in } !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle \, \underline{1}$$

where $P \stackrel{\text{def}}{=} \lambda a : \mathbb{N}.\texttt{if } \alpha\theta > \underline{5} \texttt{ then error else } a.$

$$\begin{array}{lll} & !_{\mathfrak{r}(r)} \text{ let } u =!^{\alpha}_{e_1} P \text{ in } !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle \, \underline{1} & \\ \mapsto_{\beta_\Box} & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))} \; !^{\gamma}_{\mathfrak{r}(t)} (e_1 \rhd P)\{\alpha/\gamma\} \, \underline{1} & \text{Trail persistence} \\ = & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))} \; !^{\gamma}_{\mathfrak{r}(t)} (e_1\{\alpha/\gamma\} \rhd P\{\alpha/\gamma\}) \, \underline{1} & \text{Distribute subst.} \end{array}$$

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)}\ \text{let}\ u = !^{\alpha}_{e_1} P\ \text{in}\ !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\, \underline{1}$$

where $P \stackrel{\text{def}}{=} \lambda a : \mathbb{N}.\mathtt{if}\ \alpha\theta > \underline{5}\ \mathtt{then}\ \mathtt{error}\ \mathtt{else}\ a$.

$$\begin{array}{ll}
 & !_{\mathfrak{r}(r)}\ \text{let}\ u = !^{\alpha}_{e_1} P\ \text{in}\ !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\, \underline{1} \\
\mapsto_{\beta_\Box} & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\ !^{\gamma}_{\mathfrak{r}(t)}(e_1 \rhd P)\{\alpha/\gamma\}\, \underline{1} \\
= & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\ !^{\gamma}_{\mathfrak{r}(t)}(e_1\{\alpha/\gamma\} \rhd P\{\alpha/\gamma\})\, \underline{1} \\
= & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\ !^{\gamma}_{\mathfrak{t}(e_2,\mathfrak{r}(t))} P\{\alpha/\gamma\}\, \underline{1}
\end{array}$$

Trail persistence

Distribute subst.

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)}\ \mathsf{let}\ u = !^{\alpha}_{e_1} P\ \mathsf{in}\ !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\ \underline{1}$$

where $P \stackrel{\text{def}}{=} \lambda a : \mathbb{N}.\mathtt{if}\ \alpha\theta > \underline{5}\ \mathtt{then\ error\ else}\ a$.

$$
\begin{array}{lll}
 & !_{\mathfrak{r}(r)}\ \mathsf{let}\ u = !^{\alpha}_{e_1} P\ \mathsf{in}\ !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\ \underline{1} & \\
\mapsto_{\beta_\Box} & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\ !^{\gamma}_{\mathfrak{r}(t)}(\boxed{e_1} \rhd P)\{\alpha/\gamma\}\ \underline{1} & \text{Trail persistence} \\
= & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\ !^{\gamma}_{\mathfrak{r}(t)}(e_1\{\alpha/\gamma\} \rhd P\{\alpha/\gamma\})\ \underline{1} & \text{Distribute subst.} \\
= & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\ !^{\gamma}_{\boxed{\mathfrak{t}(e_2,\mathfrak{r}(t))}} P\{\alpha/\gamma\}\ \underline{1} & \text{Trail normalisation}
\end{array}
$$

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)} \,\mathsf{let}\, u = !^{\alpha}_{e_1} P \,\mathsf{in}\, !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle \underline{1}$$

where $P \stackrel{\text{def}}{=} \lambda a : \mathbb{N}.\mathtt{if}\ \alpha\theta > \underline{5}\ \mathtt{then\ error\ else}\ a$.

$$
\begin{array}{ll}
 & !_{\mathfrak{r}(r)} \,\mathsf{let}\, u = !^{\alpha}_{e_1} P \,\mathsf{in}\, !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle \underline{1} \\
\mapsto_{\beta_\Box} & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))} \; !^{\gamma}_{\mathfrak{r}(t)} (\boxed{e_1} \rhd P)\{\alpha/\gamma\} \underline{1} \\
= & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))} \; !^{\gamma}_{\mathfrak{r}(t)} (e_1\{\alpha/\gamma\} \rhd P\{\alpha/\gamma\}) \underline{1} \\
= & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))} \; !^{\gamma}_{\boxed{\mathfrak{t}(e_2,\mathfrak{r}(t))}} P\{\alpha/\gamma\} \underline{1} \\
= & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))} \\
 & \quad !^{\gamma}_{\mathfrak{t}(e_2,\mathfrak{r}(t))} (\lambda a : \mathbb{N}.\mathtt{if}\ \gamma\theta > \underline{5}\ \mathtt{then\ error\ else}\ a) \underline{1}
\end{array}
$$

Trail persistence

Distribute subst.

Trail normalisation

Sample Reduction Step in $\lambda^{\mathcal{H}}$

$$!_{\mathfrak{r}(r)} \text{ let } u = !^{\alpha}_{e_1} P \text{ in } !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\, \underline{1}$$

where $P \stackrel{\text{def}}{=} \lambda a : \mathbb{N}.\texttt{if } \alpha\theta > \underline{5} \texttt{ then error else } a$.

$$\begin{array}{ll} & !_{\mathfrak{r}(r)} \text{ let } u = !^{\alpha}_{e_1} P \text{ in } !^{\gamma}_{\mathfrak{r}(t)} \langle u; \{\alpha/\gamma\}\rangle\, \underline{1} \\ \mapsto_{\beta_\Box} & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\; !^{\gamma}_{\mathfrak{r}(t)} (e_1 \rhd P)\{\alpha/\gamma\}\, \underline{1} \\ = & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\; !^{\gamma}_{\mathfrak{r}(t)} (e_1\{\alpha/\gamma\} \rhd P\{\alpha/\gamma\})\, \underline{1} \\ = & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))}\; !^{\gamma}_{\mathfrak{t}(e_2,\mathfrak{r}(t))} P\{\alpha/\gamma\}\, \underline{1} \\ = & !_{\mathfrak{t}(\mathfrak{bb}(u^{A[\Sigma]}.t,\Sigma.s),\mathfrak{r}(r))} \\ & \quad !^{\gamma}_{\mathfrak{t}(e_2,\mathfrak{r}(t))} (\lambda a : \mathbb{N}.\texttt{if } \gamma\theta > \underline{5} \texttt{ then error else } a)\, \underline{1} \end{array}$$

Trail persistence

Distribute subst.

Trail normalisation

Rewiring of trail vars

History-Based Access Control [AbadiFournet:2003]

If $\mathbf{f} \doteq !^{\vec{\alpha}}_{e}\lambda\vec{a} : \vec{A}.M$, then $\mathbf{f}\ \vec{\beta}\ \vec{N}$ abbreviates $\mathsf{let}\ u = \mathbf{f}\ \mathsf{in}\ \langle u; \vec{\alpha}/\vec{\beta}\rangle\ \vec{N}$

$$
\begin{array}{lcl}
\mathbf{delete} & \doteq & !^{\alpha_d}_{\mathfrak{r}(q)}\lambda a.\mathtt{if}\ \mathit{FileIOPerm} \in \theta\alpha_d\ \mathtt{then}\ \mathbf{Win32Delete}\ a \\
 & & \qquad\qquad\qquad\qquad \mathtt{else}\ \mathbf{securityException}; \\
\mathbf{cleanup} & \doteq & !^{\alpha_c}_{\mathfrak{r}(r)}\lambda a.\mathbf{delete}\ \alpha_c\ a; \\
\mathbf{bad} & \doteq & !^{\alpha_b}_{\mathfrak{r}(s)}\mathbf{cleanup}\ \alpha_b\ \text{``}..\backslash passwd'';
\end{array}
$$

θ (sample cases)

$$
\begin{array}{rcl}
\theta(\mathfrak{r}) & \stackrel{\text{def}}{=} & \emptyset \\
\theta(\mathfrak{App}) & \stackrel{\text{def}}{=} & \lambda a : \mathbb{N}.\lambda b : \mathbb{N}.a \cap b \\
\theta(\beta) & \stackrel{\text{def}}{=} & \emptyset \\
\theta(\delta_{\mathbf{f}}) & \stackrel{\text{def}}{=} & \{\mathit{perms}(\mathbf{f})\}
\end{array}
\qquad
\begin{array}{rcl}
\mathit{perms}(\mathbf{bad}) & \stackrel{\text{def}}{=} & \emptyset \\
\mathit{perms}(\mathbf{cleanup}) & \stackrel{\text{def}}{=} & \{\mathit{FileIOPerm}\} \\
\mathit{perms}(\mathbf{delete}) & \stackrel{\text{def}}{=} & \{\mathit{FileIOPerm}\}
\end{array}
$$

Evaluation of $!^{\alpha}_{\mathfrak{r}(s)}\mathbf{bad}\ \alpha$ produces security exception: $\delta_{\mathbf{bad}}(s')$ (s' unspecified) occurs in the trail consulted by $\mathbf{delete}$

1 Justification Logic

2 Hypothetical Justification Logic

3 History-Aware Computation

4 The Certifying Mobile Calculus

5 Conclusions and avenues for further research

Mobile = Absence of local dependencies

$$x + y :: \mathsf{Int}$$

- Value of $x + y$ depends on x and y
  - x and y are local resources
- Hence, $x + y$ is not mobile

Closed terms is poor approximation

$$\lambda x.\lambda y.x + y :: \mathsf{Int} \to \mathsf{Int} \to \mathsf{Int}$$

- Could be catalogued as mobile
- This state of affairs is not reflected in the type
  - Type-based static analysis for mobility?
- Restrictive
  - Mobile code could refer to other (unknown) mobile code components

Term constructor for mobile code

$$box\ (\lambda x.\lambda y.x + y) :: \Box\,(\mathsf{Int} \to \mathsf{Int} \to \mathsf{Int})$$

- box term constructor for introducing mobile code
- Mobility reflected in type
- Type-based analysis of mobility
  - Static guarantee of condition of mobility
  - Correct composition of mobile code

$$\frac{\Delta; \cdot \vdash M : A}{\Delta; \Gamma \vdash box\ M : \Box A}\ \Box\mathsf{I}$$

A more interesting example

$$\begin{array}{l} unpack\ x\ as\ u\ in \\ \quad unpack\ y\ as\ v\ in \\ \quad\quad box\ (u\,v) :: \Box\,\mathsf{Int} \end{array}$$

- Constructing mobile code out of other mobile components
  1. Extract function from mobile code x (call it u)
  2. Extract argument from mobile code y (call it v)
  3. Build application of u to v (do not execute!)
  4. Reify as mobile code

Mobility through the Justification Logic looking glass

- $\Box$ already identified to satisfy axioms of **IS4/IS5**
  - Moody, Harper, Crary, Pfenning, Murphy, Walker, etc.
  - **IS4** interpretation: reasoning about code that can execute anywhere (no explicit world indicated)
  - **IS5** interpretation: includes explicit world reference and movement primitives
- $\mathbf{JL}$ refines $\mathbf{S4}$
  - $\Box A$ replaced by $s{::}A$ ("s *is a certificate of validity for* A")

Mobility through the Justification Logic looking glass

- $\Box$ already identified to satisfy axioms of **IS4/IS5**
  - Moody, Harper, Crary, Pfenning, Murphy, Walker, etc.
  - **IS4** interpretation: reasoning about code that can execute anywhere (no explicit world indicated)
  - **IS5** interpretation: includes explicit world reference and movement primitives
- $\mathbf{JL}$ refines $\mathbf{S4}$
  - $\Box A$ replaced by $s{::}A$ ("s *is a certificate of validity for* A")

What does $\mathbf{JL}$ add to mobility?

The Certifying Mobile Calculus – $\lambda^{\mathsf{Cert}}_{\Box}$

- Replace mobile code with mobile units: $box_s\, M$
  - M is the code component
  - s is the certificate component
- Certificates
  - encode type derivations
  - are part of the object-language ($box_s\, M$ has a certificate too)
- Sample code

$$\begin{array}{l} \lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ}\rangle in \\ \quad unpack\ b\ to\ \langle v^{\bullet}, v^{\circ}\rangle\ in \\ \quad (box_{u^{\circ}\cdot v^{\circ}}\ u^{\bullet}v^{\bullet}) \end{array}$$

How are certificates used?

- How are certificates constructed?
  - Type system ensures statically that all generated certificates are valid
- When are they checked?
  - Type system ensures statically correct certificate/code correspondence
  - They are not checked at run-time
- The Certifying Mobile Calculus as a framework for certificate construction and mobile code certification

Typing Judgements

- Mobile units assumed to exist at certain worlds

$$\Delta ; \Gamma \rhd M : A@w \mid s$$

Typing Judgements

- Mobile units assumed to exist at certain worlds
- Local values extant at specific worlds

$$\Delta ; \Gamma \rhd M : A@w \mid s$$

Typing Judgements

- Mobile units assumed to exist at certain worlds
- Local values extant at specific worlds

$$\Delta ; \Gamma \rhd M : A@w \mid s$$

- Current term being typed at world w

Typing Judgements

- Mobile units assumed to exist at certain worlds
- Local values extant at specific worlds

$$\Delta ; \Gamma \rhd M : A@w \mid s$$

- Current term being typed at world w
- Current certificate

Sample Typing Schemes

$$\frac{}{\Delta, v : A@w, \Delta'; \Gamma \rhd v^{\bullet} : A@w \mid v^{\circ}} \text{VarV}$$

$$\frac{\Delta; \cdot \rhd M : A@w \mid s}{\Delta; \Gamma \rhd box_s\, M : [s]A@w \mid !s} \Box I$$

$$\frac{\Delta; \Gamma \rhd M : [r]A@w \mid s \quad \Delta, v : A@w; \Gamma \rhd N : C@w \mid t}{\Delta; \Gamma \rhd unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N : C_r^{v^{\circ}}@w \mid letc\ s\ be\ v : A\ in\ t} \Box E$$

Reduction Judgements

- Network: set of worlds with their corresp. continuation stacks

$$\mathbb{W}\, ;\, w : [\, k\, ,\, M\,]$$

Reduction Judgements

- Network: set of worlds with their corresp. continuation stacks
- Node where computation is currently taking place

$$\mathbb{W} ; w : [k, M]$$

Reduction Judgements

- Network: set of worlds with their corresp. continuation stacks
- Node where computation is currently taking place

$$\mathbb{W} ; w : [k , M]$$

- Current continuation

Reduction Judgements

- Network: set of worlds with their corresp. continuation stacks
- Node where computation is currently taking place

$$\mathbb{W}; w : [k, M]$$

- Current continuation
- Current focus of computation

Dynamics: Abstract machine reduction over multiple nodes

$$\begin{array}{rcl} \mathbb{W}; w : [k, M\ N] & \rightarrow & \mathbb{W}; w : [k \triangleleft \circ\ N, M] \\ \mathbb{W}; w : [k \triangleleft \circ\ N, V] & \rightarrow & \mathbb{W}; w : [k \triangleleft V\ \circ, N] \\ \mathbb{W}; w : [k \triangleleft (\lambda a.M)\ \circ, V] & \rightarrow & \mathbb{W}; w : [k, M^a_V] \end{array}$$

$$\begin{array}{c} \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] \rightarrow \\ \mathbb{W}; w : [k \triangleleft \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] \\ \mathbb{W}; w : [k \triangleleft \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] \rightarrow \\ \mathbb{W}; w : [k, (N^{v^\circ}_s)^{v^\bullet}_M] \end{array}$$

$$\begin{array}{rcl} \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] & \rightarrow & \{w : C :\ : k; w_s\}; w' : [\mathit{return}\ w, M] \\ \{w : C :\ : k; w_s\}; w' : [\mathit{return}\ w, V] & \rightarrow & \{w : C; w_s\}; w : [k, V^{w'}_w] \end{array}$$

Results

- Relation to $\mathbf{JL}^m$

  If $\Delta;\Gamma \rhd M : A@w|s$ is derivable, then $\Delta';\Gamma' \vdash A' \mid s'$ (obtained basically by erasing location qualifiers from the latter) is derivable in $\mathbf{JL}^m$

- Subject reduction ($\mathbb{N}$ machine state)

  If $\Sigma \vdash \mathbb{N}$ is derivable and $\mathbb{N} \to \mathbb{N}'$, then $\Sigma \vdash \mathbb{N}'$ is derivable

- Strong normalization of typable states (by reduction to SN of $\lambda^{\to}$)

  All typable machine states are strongly normalizing

1 Justification Logic

2 Hypothetical Justification Logic

3 History-Aware Computation

4 The Certifying Mobile Calculus

5 Conclusions and avenues for further research

Summing up

- Overview of $\mathbf{JL}$
- Natural deduction for $\mathbf{JL}^m$
- Explored two possible computational interpretations
  - History-based computation ($\lambda^{\mathcal{H}}$)
  - Certifying mobile calculus ($\lambda^{\mathsf{Cert}}_{\Box}$)

$\lambda^{\mathcal{H}}$ (1/2)

- History aware lambda calculus, $\lambda^{\mathcal{H}}$
- Fine operations on audited units obtained by purely logical means
  - Trail persistence
  - Trail variable rewiring
- Properties of $\lambda^{\mathcal{H}}$
  - Safety: Subject Reduction+Progress
  - SN (for a restriction)
- Other examples
  - Abadi & Fournet'03: History-based access control
  - Banerjee & Naumann'04: History-based a.c. for information flow

$\lambda^{\mathcal{H}}$ (2/2)

- Model more examples from security domain
- Affine nature of trail variables imposes similar restriction to term variables
- Expressiveness
  - Cannot jump to previous point in trail
- Explore programming idiom
  - Modal term constructor introduces local scope for trail variables
  - Trails maintained by the run-time system
  - Trails can be discarded

$$\text{let } u = !^{\Sigma}_{e} M \text{ in } !^{\Sigma'}_{e'} N \text{ with } u \notin \text{fmv}(N)$$

$\lambda^{\mathsf{Cert}}_{\Box}$

- Unified framework for certificate and mobile code construction
- Provides static guarantees for both
  1. Non-dependency on local resources (mobility)
  2. Valid formation of compound certificates
  3. Correct certificate/code correspondence
- From a programming languages perspective, just a teaser
  - Must at least make available:
    1. Imperative features (eg. references)
    2. Fixed point computation
    3. Non-local references ($\diamond A$)
    4. Certificate "polymorphism"

$$\Lambda x.\Lambda y.x{::}(A \supset B) \supset y{::}A \supset x \cdot y{::}B$$

Questions?